

```
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AAAAAA MM MM PPPPPPPP LL EEEEEEEEEE SSSSSSSS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MMMM MMMM PP PP LL EE SS
EE XX XX AA AA MM MM PP PP LL EE SS
EE XX XX AA AA MM MM PP PP LL EE SS
EEEEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSS
EEEEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSS
EEEEEEEEEE XX XX AA AA MM MM PPPPPPPP LL EEEEEEEEE SSSSSS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AAAAAAAAAA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EE XX XX AA AA MM MM PP LL EE SS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
EEEEEEEEEE XX XX AA AA MM MM PP LLLLLLLLLL EEEEEEEEE SSSSSSSS
```

```

      AAAAAA      DDDDDDDD      DDDDDDDD      RRRRRRRR      IIIIII      VV      VV      EEEEEEEEEE      RRRRRRRR
      AAAAAA      DDDDDDDD      DDDDDDDD      RRRRRRRR      IIIIII      VV      VV      EEEEEEEEEE      RRRRRRRR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AAAAAAAAAA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AAAAAAAAAA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DD      DD      DD      DD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DDDDDDDD      DDDDDDDD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR
AA      AA      DDDDDDDD      DDDDDDDD      RR      RR      RR      RR      RR      RR      RR      RR      RR      RR

```

```

MM      MM      AAAAAA      RRRRRRRR
MM      MM      AAAAAA      RRRRRRRR
MMMM      MMMM      AA      AA      RR      RR      RR
MMMM      MMMM      AA      AA      RR      RR      RR
MM      MM      MM      AA      AA      RR      RR      RR
MM      MM      MM      AA      AA      RRRRRRRR
MM      MM      MM      AA      AA      RRRRRRRR
MM      MM      AAAAAAAAAA      RR      RR
MM      MM      AAAAAAAAAA      RR      RR
MM      MM      AA      AA      RR      RR      RR
MM      MM      AA      AA      RR      RR      RR
MM      MM      AA      AA      RR      RR      RR
MM      MM      AA      AA      RR      RR      RR

```

ADDR

: CO
: US
:

AD_E

: CS
:
AD_C

: DE
: PC
:

AD_1

: PC
: LC
: DU
:

AD_F
10\$

: NC
:
AD_I

.TITLE ADDRIVER - VAX/VMS AD11-K DRIVER
.IDENT 'V04-000'

*
* COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
* DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
* ALL RIGHTS RESERVED.
*

* THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
* ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
* INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
* COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
* OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
* TRANSFERRED.
*

* THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
* AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
* CORPORATION.
*

* DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
* SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
*

++

FACILITY:

VAX/VMS AD11-K I/O DRIVER

ABSTRACT:

DEVICE TABLES AND DRIVER CODE FOR THE AD11-K ANALOGUE
TO DIGITAL CONVERTER WITH OPTIONAL AM11-K MULTIPLEXER.

AUTHOR:

S. PROGRAMMER, SEPTEMBER 1978.

MODIFIED BY:

--

.SBTTL FUNCTIONAL DESCRIPTION OF DRIVER

THE DRIVER SUPPORTS A/D SAMPLING ON GROUPS OF CHANNELS VIA QIO READ REQUESTS. NO EXTERNALLY TRIGGERED SAMPLING (I.E., CLOCK OVERFLOW OR SCHMITT TRIGGER) IS SUPPORTED. THE AM11-K MULTIPLEXER MAY BE PRESENT, BUT NO AUTOMATIC RANGING AMPLIFICATION IS DONE AT DRIVER LEVEL. THE BUILT-IN DAC MAY BE USED FOR TESTING VIA A LOOPBACK QIO FUNCTION DEFINED ESPECIALLY FOR THIS DEVICE.

THE QIO FUNCTIONS AVAILABLE ARE:

IOS_READVBLK	-READ VIRTUAL BLOCK
IOS_READLBLK	-READ LOGICAL BLOCK
IOS_READPBLK	-READ PHYSICAL BLOCK=IOS_LOOPBACK
IOS_LOOPBACK	-WRITE DAC, READ RESULTS; REQUIRES PHYSICAL I/O PRIVILEGE

THE STANDARD QIO PARAMETERS ARE:

P1=BUFFER ADDRESS
P2=BUFFER BYTE COUNT
P3=SPECIFIER OF CHANNELS TO SAMPLE:
 BIT 0-7/INITIAL CHANNEL # (0-63)
 BIT 8-15/TOTAL # OF CHANNELS TO SAMPLE (1-64)
 BIT 16-23/CHANNEL INCREMENT (0-63)
 BIT 24-31/IGNORED
P4=DAC VALUE, USED FOR LOOPBACK ONLY:
 BIT 0-7/8 BIT DAC VALUE
 BIT 8-31/IGNORED
P5,P6 ARE NOT USED

IN ADDITION TO THE STANDARD STATUS CODES THAT CAN BE RETURNED FOR A QIO, THE FOLLOWING DEVICE-SPECIFIC I/O STATUS VALUES ARE DEFINED:

SS\$_DATAOVERUN	-ERROR BIT SET IN CSR; SAMPLING ABORTED WITH LAST GOOD SAMPLE IN BUFFER
SS\$_BADPARAM	-INVALID CHANNEL SPECIFIER; NO SAMPLES TAKEN
SS\$_BUFFEROVF	-USER BUFFER OVERRUN; AS MANY CHANNELS AS WILL FIT ARE SAMPLED

THE SAMPLES ARE RETURNED IN THE CALLER'S BUFFER PACKED ONE SAMPLE PER WORD, BITS 0-11. THE BYTE COUNT RETURNED IN THE SECOND WORD OF THE I/O STATUS BLOCK ALWAYS REFLECTS THE # OF BYTES ACTUALLY FILLED WITH SAMPLE DATA. THE NUMBER OF SAMPLES IS ONE HALF THE RETURNED BYTE COUNT.

EXAMPLE: SWEEP THROUGH 32 INPUTS CONNECTED IN DIFFERENTIAL MODE (AD11-K AND AM11-K):

SWEEPBUF:	.BLKW	32	
NUMINPUT:	.LONG	32	
CHANSPEC:	.BYTE	0,32,2	; START WITH CHANNEL 0; ; SAMPLE CHANNELS 0,2,4,....,62

\$QIO_S CHAN=X, FUNC=IOS_READVBLK, -
P1=SWEEPBUF, P2=NUMINPUT, P3=CHANSPEC

ADDRIVER.MAR;1

16-SEP-1984 17:04:05.93 Page 3

;-

ADDI

+
AD
TH
F
TH
L
IS
SE
TH
II
II
ON
-
AD_

.SBTTL MACRO LIBRARY CALLS

: EXTERNAL SYMBOLS (LIB/LIB):

\$CRBDEF	: CHANNEL REQUEST BLOCK
\$DDBDEF	: DEVICE DATA BLOCK
\$IDBDEF	: INTERRUPT DATA BLOCK
\$IODEF	: I/O FUNCTION CODES
\$IPLDEF	: HARDWARE IP DEFINITIONS
\$IRPDEF	: I/O REQUEST PACKET
\$UCBDEF	: UNIT CONTROL BLOCK
\$VECDEF	: INTERRUPT VECTOR BLOCK
\$JIBDEF	: JOB INFORMATION BLOCK

: USER DEFINED EXTERNAL SYMBOLS ARE CONTAINED IN A USER LIBRARY .
: THE CONTENTS OF THIS LIBRARY CAN BE MERGED WITH THE SYSTEM LIBRARY
: TO ALLOW USER PROGRAMS TO USE EXTENDED FUNCTION CODES WITHOUT HAVING
: TO DEFINE THEM LOCALLY.
: THIS DRIVER MUST BE ASSEMBLED WITH A USER LIBRARY TO DEFINE \$XIODEF.

\$XIODEF	: EXTENDED QIO FUNCTIONS. THIS MACRO
	: CONTAINS THE DEFINITIONS FOR
	: IOS_LOOPBACK

.SBTTL LOCAL DEFINITIONS

```
;; LOCAL DEFINITIONS:
;; QIO ARGUMENT LIST OFFSETS:
;;
```

```
P1=0
P2=4
P3=8
P4=12
P5=16
P6=20
```

```
; FIRST,
; SECOND,
; THIRD,
; FOURTH,
; FIFTH,
; AND SIXTH PARAMETERS
```

```
;; DEVICE PARAMETERS:
;;
```

```
DAC_TIMER=20
MAX_INLCHN=63
MAX_NUMCHN=64
MAX_INCCHN=63
ADC_TIMER=2
```

```
; 20 USEC TIMER FOR DAC SETTLE
; MAXIMUM INITIAL CHANNEL #,
; NUMBER OF CHANNELS,
; AND CHANNEL INCREMENT
; A/D CONVERSION TIMEOUT=2 SEC
```

```
;; DEVICE REGISTER DEFINITIONS:
;;
```

```
      $DEFINI AD
$DEF  AD_CSR .BLKW 1
      VIELD AD_CSR,0,<-
      ZGO,,M>,-
      <,3>,-
      <EXT,,M>,-
      <COV,,M>,-
      <IE,,M>,-
      <DON,,M>,-
      <MUX,6,M>,-
      <,1>,-
      <ERR,,M>,-
      >
$DEF  AD_DBR .BLKW 1
      =-2
$DEF  AD_DAC .BLKW 1
      $DEFEND AD
```

```
; CONTROL/STATUS REGISTER
; DEFINE CSR FIELDS: AD_CSR_M_XXX
; START A/D CONVERSION
; 3 UNUSED BITS
; EXTERNAL START ENABLE
; CLOCK OVERFLOW ENABLE
; INTERRUPT ENABLE
; CONVERSION DONE FLAG
; 6 BIT MUX CHANNEL #
; BIT 14 IS UNUSED
; ERROR FLAG
; END OF CSR FIELDS
; A/D DATA BUFFER REGISTER
; DATA BUFF REG=DAC BUFF REG
; DAC DATA BUFFER REF
; END OF A/D REGISTER DEFNS
```

```
;; DEVICE DEPENDENT UCB EXTENSIONS:
;;
```

```
$DEFINI UCB
```

```
DDI
DDI
DDI
DDI
DDI
DDI
DDI
DDI
DDI
DDI
DDI
DDI
```

```
MM
MM
MM
MM
MM
MM
MM
MM
MM
MM
MM
MM
```

```

.=UCB$K_LENGTH                                ;STEP TO END OF STANDARD UCB
                                                ;NOTE: NEXT 4 BYTES ASSUMED
                                                ; ADJACENT
$DEF      UCB$B_AD_CURCHN .BLKB 1              ;CURRENT MUX CHANNEL #
$DEF      UCB$B_AD_NUMCHN .BLKB 1              ;# CHANNELS LEFT TO SAMPLE
$DEF      UCB$B_AD_INCCHN .BLKB 1              ;CHANNEL INCREMENT
                                                ;SPARE BYTE
$DEF      UCB$W_AD_CSR    .BLKW 1              ;SAVED CSR
        VIELD UCB$W_CSR,1,<-                  ;BORROW UNUSED CSR BIT
        ZBFO,,M>,-                             ; FOR USER BUFFER OVERRUN
    >
UCB$K_ADLENGTH=.                               ;LENGTH OF A/D UCB
        $DEFEND UCB                             ;END OF UCB EXTENSIONS
;
; A/D DRIVER USE OF TEMPORARY IRP STORAGE:
;
IRPSL_CHSPEC=IRPSL_MEDIA                       ;CHANNEL SPECIFIER(P3)
IRPSL_DACVAL=IRPSL_MEDIA+4                     ;OPTIONAL DAC VALUE(P4)
    
```


.SBTTL DRIVER PROLOGUE AND DISPATCH TABLES

: DRIVER PROLOGUE TABLE:

```

DPTAB - ;DEFINE DRIVER PROLOGUE TABLE:
      END=AD END,- ; END OF DRIVER,
      ADAPTER=UBA,- ; UNIBUS ADAPTER,
      UCBSIZE=UCBSK_AD LENGTH,- ; SIZE OF A/D UCB,
      NAME=ADDRIVER ; DRIVER NAME

DPT_STORE INIT ;VALUES TO BE SET ON LOAD
DPT_STORE UCB,UCBSB_FIPL,B,8 ;DEVICE FORK IPL
DPT_STORE UCB,UCBSB_DIPL,B,22 ;AD11 HARDWARE IPL
DPT_STORE UCB,UCBSL_DEVCHAR,L,- ;AD11 DEVICE CHARACTERISTICS:
      <DEVS_M_AVC- ; AVAILABLE,
      !DEVS_M_IDV- ; INPUT DEVICE,
      !DEVS_M_RTM> ; REALTIME DEVICE

DPT_STORE REINIT ;VALUES TO SET ON RELOAD
DPT_STORE CRB,CRBSL_INTD+4,D,- ;INTERRUPT SERVICE ADDR
      AD_INTERRUPT
DPT_STORE CRB,- ;ADDR OF CONTROLLER
      CRBSL_INTD+VECSL_INITIAL,- ; INITIALIZATION
      D,AD_CTLINIT
DPT_STORE CRB,- ;ADDR OF UNIT
      CRBSL_INTD+VECSL_UNITINIT,- ; INITIALIZATION
      D,AD_UNITINIT
DPT_STORE DDB,DBBSL_DDT,D,- ;ADDR OF DRIVER
      AD$DDT ; DISPATCH TABLE

DPT_STORE END ;END DRIVER PROLOGUE

```

: DRIVER DISPATCH TABLE:

```

DDTAB - ;DDT CREATION MACRO
      DEVNAM=AD,- ;NAME OF DEVICE
      START=AD_STARTIO,- ;ADDR OF START I/O ROUTINE
      FUNCTB=AD_FUNCABLE ;ADDR OF FDT

```

.SBTTL AD11-K FUNCTION DECISION TABLE

:
: AD11 FUNCTION DECISION TABLE:
:

AD_FUNCTABLE:		:FUNCTION DECISION TABLE START
FUNCTAB	,-	:LEGAL FUNCTIONS:
	<LOOPBACK,-	: LOOPBACK READ FROM DAC
	READPBLK,-	: READ PHYSICAL BLOCK
	READLBLK,-	: READ LOGICAL BLOCK
	READVBLK>	: READ VIRTUAL BLOCK
FUNCTAB	,-	:BUFFERED I/O FUNCTIONS:
	<LOOPBACK,-	: LOOPBACK READ FROM DAC
	READPBLK,-	: READ PHYSICAL BLOCK
	READLBLK,-	: READ LOGICAL BLOCK
	READVBLK>	: READ VIRTUAL BLOCK
FUNCTAB	-	:PREPROCESSING ROUTINES:
	AD_READ,-	:CALL SINGLE PREPROCESSOR FOR:
	<LOOPBACK,-	: LOOPBACK READ FROM DAC
	READPBLK,-	: READ PHYSICAL BLOCK
	READLBLK,-	: READ LOGICAL BLOCK
	READVBLK>	: AND READ VIRTUAL BLOCK

.SBTTL AD_READ: READ FUNCTION PROCESSING

AD_READ - READ FUNCTION PREPROCESSING

THIS ROUTINE IS CALLED FROM THE FUNCTION DECISION TABLE DISPATCHER TO PROCESS A READ PHYSICAL, READ LOGICAL, READ VIRTUAL, OR LOOPBACK I/O FUNCTION.

AD_READ FIRST VERIFIES THE CALLER'S PARAMETERS, TERMINATING THE REQUEST WITH IMMEDIATE SUCCESS OR ERROR IF NECESSARY. P3 AND P4 ARE STORED IN THE IRP. A SYSTEM BUFFER IS ALLOCATED AND ITS ADDRESS IS SAVED IN THE IRP. THE CALLER'S QUOTA IS UPDATED, AND THE READ REQUEST IS QUEUED TO THE DRIVER FOR STARTUP.

INPUTS:

R0,R1,R2 = SCRATCH
R3 = IRP ADDRESS
R4 = ADDR OF PCB FOR CURRENT PROCESS
R5 = DEVICE UCB ADDRESS
R6 = ADDRESS OF CCB
R7 = I/O FUNCTION CODE
R8 = FDT DISPATCH ADDR
R9-R11 = SCRATCH
AP = ADDR OF FUNCTION PARAMETER LIST

OUTPUTS:

R0,R1,R2 = DESTROYED
R3-R11,AP = PRESERVED
IRP\$\$_CHSPEC(R3) = CHANNEL SPECIFIER (P3)
IRP\$\$_DACVAL(R3) = OPTIONAL DAC VALUE (P4)
IRP\$\$_SVAPTE(R3) = ADDR OF ALLOCATED SYSTEM BUFFER
IRP\$\$_BOFF(R3) = REQUESTED BYTE COUNT

SYSTEM BUFFER:

LONGWD 0/ADDR OF START OF DATA=BUFF ADDR+12
LONGWD 1/ADDR OF USER BUFFER
LONGWD 2/DATA STRUCTURE BOOKKEEPING

.ENABL LSB

AD_READ:		:READ FUNCTION PREPROCESSING
MOVZWL	P2(AP),R1	:GET USER BYTE COUNT
BEQL	10\$	:BRANCH IF READ OF 0 BYTES
		: (=INSTANT SUCCESS)
MOVZWL	#SS\$ BADPARAM,R0	:ASSUME CHANNEL SPEC ERROR
MOVAL	P3(AP),R2	:GET ADDR OF CHANNEL SPEC
CMPB	(R2)+,#MAX_INLCHN	:INITIAL CHAN # TOO LARGE?
BGTRU	20\$	:BRANCH IF SO
TSTB	(R2)	:# CHANNELS = 0?
BEQL	10\$	:BRANCH IF SO (SUCCESS)
CMPB	(R2)+,#MAX_NUMCHN	:# CHANNELS TO SAMPLE TOO LARGE?
BGTRU	20\$	:BRANCH IF SO
CMPB	(R2),#MAX_INCCHN	:CHANNEL INCREMENT TOO LARGE?

```

BGTRU 20$          ;BRANCH IF SO
MOVQ   P3(AP),IRP$L_CHSPEC(R3) ;STORE P3 AND P4 (OPTIONAL DAC)
; IN IRP UNTIL REQUEST EXECUTION
MOVL   P1(AP),R0    ;GET ADDR OF USER BUFFER
JSB    G^EXE$READCHK ;VERIFY THAT CALLER HAS
; WRITE ACCESS TO BUFFER
PUSHR  #^M<R0,R3>   ;SAVE USER BUFF ADDR, IRP ADDR
ADDL   #12,R1        ;ADD 12 BYTES TO REQUESTED BUFF
; SIZE FOR BUFF HEADER
JSB    G^EXE$BUFFRQUOTA ;VERIFY BUFFER SPACE LEFT
; IN CALLER'S QUOTA
BLBC   R0,30$        ;BRANCH IF INSUFFICIENT QUOTA
JSB    G^EXE$ALLOCBUF ;ALLOCATE A SYSTEM BUFFER
BLBC   R0,30$        ;BRANCH IF NONE AVAILABLE
POPR   #^M<R0,R3>   ;RESTORE USER BUFFER, IRP ADDR
MOVL   R2,IRP$L_SVAPTE(R3) ;SAVE ADDR OF SYSTEM BUFFER
MOVW   R1,IRP$W_BOFF(R3)  ; AND REQUESTED BYTE COUNT
MOVZWL R1,R1         ;CONVERT TO LONGWORD
MOVL   PCB$L_JIB(R4),R4   ;GET JOB INFORMATION BLOCK ADDRESS
SUBL   R1,JIB$L_BYTCNT(R4) ;DEDUCT REQUESTED BYTE COUNT
; FROM PROCESS' QUOTA
MOVAB  12(R2),(R2)+      ;SAVE ADDR OF START OF USER DATA
; IN 1ST LONGWD OF SYSTEM BUFFER
MOVL   R0,(R2)          ;SAVE USER BUFFER ADDR IN
; 2ND LONGWD
JMP    G^EXE$QIODRVPKT  ;QUEUE I/O PKT TO DRIVER

```

```

; COME HERE IF USER REQUESTED READ OF 0 BYTES OR 0 CHANNELS.
; THIS IS ALWAYS SUCCESSFUL AND DOES NO DEVICE I/O:
;

```

```

10$:  MOVZWL  #SS$ NORMAL,R0      ;SET NORMAL COMPLETION STATUS
20$:  JMP     G^EXE$FINISHIOC     ;COMPLETE I/O REQUEST

```

```

; COME HERE TO ABORT I/O REQUEST WITH EXCEPTION STATUS IN R0:
;

```

```

30$:  POPR    #^M<R2,R3>          ;CLEAR BUFFER ADDR; RESTORE IRP
; ADDR
JMP    G^EXE$ABORTIO             ;COMPLETE I/O REQUEST
.DSABL LSB

```


.SBTTL AD_STARTIO: PERFORM A/D CONVERSIONS

AD_STARTIO - START I/O OPERATION ON AD11-K A/D CONVERTER.

THIS ROUTINE IS ENTERED WHEN THE ASSOCIATED UNIT IS IDLE AND A PACKET IS AVAILABLE FOR PROCESSING.

TO PREPARE FOR SAMPLING, AD_STARTIO PERFORMS THESE STEPS:

1. SET UP UCB WITH CHANNEL SPECIFIER AND ADDRESS IN SYSTEM BUFFER TO HOLD FIRST SAMPLE.
2. IF LOOPBACK WAS SPECIFIED, THE DAC IS SET WITH THE CALLER-SPECIFIED VALUE.

THE DRIVER THEN LOOPS FROM AD NXTSAMPLE TO AD ENDSAMPLE COLLECTING SAMPLES UNTIL ALL SAMPLES HAVE BEEN COLLECTED, OR AN ERROR OCCURS. AN INTERRUPT IS RECEIVED FOR EACH SAMPLE, BUT, TO SAVE TIME, THE DRIVER NEVER FORKS UNTIL TIME TO COMPLETE THE I/O REQUEST.

INPUTS:

R3 = ADDR OF IRP
R5 = ADDR OF DEVICE UNIT UCB

OUTPUTS:

R0,R1,R2 = DESTROYED
OTHER REGISTERS ARE PRESERVED

.ENABL LSB

```

AD_STARTIO:                                ;START NEXT QIO
      MOVL  IRPSL_CHSPEC(R3),-              ;COPY CHANNEL SPEC FROM
      UCB$B_AD_CURCHN(R5)                  ; IRP TO UCB
      MOVL  @IRPSL_SWAPTE(R3),-            ;SET ADDR OF START DATA
      UCB$B_SWAPTE(R5)                    ; IN UCB
      MOVL  UCB$B_CRB(R5),R4                ;GET CRB ADDRESS,
      MOVL  @CRB$C_INTD+VEC$B_IDB(R4),R4   ; THEN CSR ADDRESS
      BICB3  #^C<IOSM FCODE>,-             ;GET THE I/O
      IRPSW_FUNC(R3),R0                   ; FUNCTION CODE
      CMPB  R0,#IOS LOOPBACK               ;LOOPBACK?
      BNEQ  AD NXTSAMPLE                   ;BRANCH IF NOT
      MOVZBW IRPSL_DACVAL(R3),-            ;SET DAC VALUE IN
      AD_DAC(R4)                          ; DAC BUFFER REGISTER
      MFPR  S^#PRS_ICR,R1                  ;GET CURRENT INTERVAL COUNTER (USEC)
      ADDL  #DAC_TIMER,R1                  ; +DAC SETTLE TIME IN USEC
      BLSS  10$                            ;BRANCH IF COUNTER DOESN'T
                                          ; OVERFLOW
      MOVAW -10000(R1),R1                  ;ELSE CALCULATE COUNTER
                                          ; FOR NEXT INTERVAL
10$:    MFPR  S^#PRS_ICR,R0                  ;READ INTERVAL COUNTER NOW
      CMPL  R0,R1                          ;REACHED SETTLE TIME YET?
      BLSS  10$                            ;BRANCH IF NOT

```



```

AD_NXTSAMPLE:      ;START NEXT SAMPLE
MOVZBW  #AD_CSR_M_IE!AD_CSR_M_GO,RO ;SET INTERRUPT ENABLE AND
; START A/D CONVERSION
INSV    UCBSB_AD_CURCHN(R5),-      ;SET MUX CHAN #
; #8,#6,RO                        ; FOR CSR
DSBINT   ;DISABLE INTERRUPTS (IPL=IPL$POWER)
BBSC     #UCBSV_POWER,-            ;BRANCH IF POWER FAILURE
; UCBSW_STS(R5),AD_POWERFAIL ; AND CLEAR POWER FAIL SIGNAL
MOVW     RO,AD_CSR(R4)             ;SET CSR
WFIKPCW  AD_TIMEOUT,#ADC_TIMER    ;WAIT FOR INTERRUPT, OR TIMEOUT
MOVW     AD_CSR(R4),UCBSW_AD_CSR(R5) ;SAVE CSR IN UCB
BLSS     AD_CSRERROR              ;BRANCH IF ERROR
MOVW     AD_DBR(R4),@UCBSL_SVAPTE(R5) ;COPY A/D VALUE INTO
; SYSTEM BUFFER
ADDL     #2,UCBSL_SVAPTE(R5)       ;STEP BUFFER POINTER
SUBL     #2,UCBSW_BCNT(R5)         ;DECREASE # BYTES LEFT IN REQUEST
DECB     UCBSB_AD_NUMCHN(R5)       ;DECR # CHANNELS LEFT TO SAMPLE
BEQL     AD_DONE                   ;BRANCH IF NONE
CMPW     UCBSW_BCNT(R5),#2         ;AT LEAST 2 BYTES LEFT IN BUFFER?
BLSSU    AD_BUFFEROVF             ;BRANCH IF NOT
BICW     #UCBSW_CSR_M_BFO,-        ;ELSE CLEAR BUFFER OVERRUN
; UCBSW_AD_CSR(R5)              ; BIT IN CSR COPY
ADDB     UCBSB_AD_INCCHN(R5),-     ;NEXT CHANNEL # =
; UCBSB_AD_CURCHN(R5)          ; CURRENT CHANNEL+INCREMENT
BICB     #^C<MAX_NUMCHN-1>,-      ; MODULO MAXIMUM
; UCBSB_AD_CURCHN(R5)          ; CHANNEL #

AD_ENDSAMPLE:      ;THIS SAMPLE COMPLETE
BRB      AD_NXTSAMPLE             ;GO START NEXT SAMPLE

.DSABL    LSB

```


.SBTTL - I/O REQUEST COMPLETION

```

: COME HERE TO COMPLETE I/O REQUEST WITH NORMAL OR ERROR STATUS.
: USER BUFFER OVERRUN, I.E., NO MORE SAMPLES CAN BE COLLECTED:

```

.ENABL LSB

```

AD_BUFFEROVF:
  BISW      #UCB$W_CSR_M_BFO,-      :SET BUFFER OVERRUN BIT
           UCB$W_AD_CSR(R5)         : IN CSR COPY

```

```

: CSR ERROR BIT WAS SET:

```

```

AD_CSRERROR:
  TSTW      AD_DBR(R4)               :CLEAR ERROR
  BRB       AD_DONE                 :JOIN COMMON I/O COMPLETION

```

```

: DEVICE TIMED OUT DUE TO EITHER A REAL TIMEOUT OR TO A
: POWER FAILURE. BOTH CAUSES ARE HANDLED THE SAME.

```

```

AD_TIMEOUT:
  CLRW      AD_CSR(R4)               :CLEAR INTERRUPT ENABLE,
  TSTW      AD_DBR(R4)               : PENDING CONVERSION, INT, OR ERROR
  SETIPL    UCB$B_FIPL(R5)           :LOWER PRIORITY TO DEVICE LEVEL
  BRB       10$                     :JOIN COMMON CODE TO
                                   : TERMINATE REQUEST

```

```

: POWER FAILURE DETECTED WHILE ATTEMPTING TO INITIATE A READ OR
: LOOPBACK REQUEST. TERMINATE REQUEST THE SAME AS IF IT OCCURRED
: DURING THE QIO.

```

```

AD_POWERFAIL:
  ENBINT
10$:  MOVZWL  #SS$_TIMEOUT,R0         :LOWER IPL BACK TO FORK IPL
      BRB    20$                     :SET STATUS TO TIMED OUT
                                   :JOIN COMMON CODE TO TERMINATE
                                   : REQUEST

```

```

: NORMAL STATUS, CANCEL I/O, AND GENERAL I/O REQUEST COMPLETION:

```

```

AD_DONE:
  CLRW      AD_CSR(R4)               :CLEAR INTERRUPT ENABLE
  IOFORK
  MOVZWL    #SS$_DATAOVRUN,R0        :REQUEST RESUMPTION AS FORK PROCESS
  BBS       #AD_CSR_V_ERR,-          :ASSUME CSR ERROR
           UCB$W_AD_CSR(R5),20$      :BRANCH IF SO

```

```
MOVZWL #SS$ BUFFEROVF,R0      ;ASSUME BUFFER OVERRUN
BBS     #UCBSW_CSR_V_BFO,-     ;BRANCH IF SO
MOVZWL  UCBSW_AD_CSR(R5),20$    ;
MOVZWL  #SS$_NORMAL,R0         ;ELSE, STATUS IS NORMAL
20$:    SUBW3 UCBSW_BCNT(R5),-    ;GET # BYTES REQUESTED
        IRPSW_BCNT(R3),R1       ; -# BYTES NOT XFERRERD
        INSV  R1,#16,#16,R0     ; =# BYTES XFERRERD
        CLRL  R1                ;CLEAR SECOND I/O STATUS LONGWD
        REQCOM                  ;REQUEST I/O COMPLETION

.DSABL  LSB
```


.SBTTL AD_INTERRUPT: AD11-K A/D CONVERTER INTERRUPT SERVICE

AD_INTERRUPT - A/D CONVERTER INTERRUPT SERVICE

THIS ROUTINE IS ENTERED VIA A JSB INSTRUCTION WHEN AN INTERRUPT OCCURS ON AN AD11 A/D CONVERTER. INTERRUPT SERVICE GETS THE ADDRESS OF THE UCB OF THE INTERRUPTING DEVICE, RESTORES THE REMAINING CONTEXT OF THE DRIVER FORK PROCESS WHICH INITIATED THE DEVICE ACTIVITY, AND CALLS THE DRIVER FORK PROCESS.

INPUTS:

ALL GENERAL REGISTERS = RANDOM
 SP/ INTERRUPT STACK
 0(SP) = ADDR OF IDB ADDR
 4(SP) = SAVED R0
 8(SP) = SAVED R1
 12(SP) = SAVED R2
 16(SP) = SAVED R3
 20(SP) = SAVED R4
 24(SP) = SAVED R5
 28(SP) = SAVED PC
 32(SP) = SAVED PSL
 IPL/ HARDWARE DEVICE LEVEL

OUTPUTS AT CALL TO DRIVER FORK:

R3 = RESTORED FROM DRIVER FORK PROCESS (IRP ADDR)
 R4 = RESTORED FROM DRIVER FORK PROCESS (CSR ADDR)
 R5 = UCB ADDR
 STACK IS SAME AS ABOVE, BUT IDB POINTER POPPED
 IPL/ HARDWARE DEVICE LEVEL

.ENABL LSB

AD_INTERRUPT:		:A/D CONVERTER INTERRUPT SERVICE
MOVL	@(SP)+,R3	:GET IDB ADDR
MOVQ	IDB\$\$(CSR(R3),R4	:GET DEVICE CSR AND UCB ADDR
BBCC	#UCB\$V_INT,-	:BRANCH IF INT UNEXPECTED,
	UCB\$W_STS(R5),AD_UN SOL	:AND CLEAR EXPECTED BIT
MOVL	UCB\$\$_FR3(R5),R3	:RESTORE REMAINING DRIVER
		:CONTEXT: R3; (R4 ALREADY SET)
JSB	@UCB\$\$_FPC(R5)	:CALL DRIVER FORK PROCESS
10\$:	MOVQ (SP)+,R0	:RESTORE REGISTERS
	MOVQ (SP)+,R2	
	MOVQ (SP)+,R4	
	REI	
AD_UN SOL:		:HANDLE UNSOLICITED INTERRUPT
CLRW	AD_CSR(R4)	:DISMISS SPURIOUS INTERRUPT
TSTW	AD_DBR(R4)	:READ DATA BUFFER TO CLEAR ERROR
BRB	10\$	:JOIN INTERRUPT RESTORE
.DSABL	LSB	

.SBTTL AD_CTLINIT: AD11-K CONTROLLER INITIALIZATION

AD_CTLINIT - AD11-K CONTROLLER INITIALIZATION

THIS ROUTINE IS CALLED AT SYSTEM STARTUP AND AFTER A POWER FAILURE.

THE CSR IS CLEARED TO DISABLE INTERRUPTS. THIS WILL FORCE THE LAST SAMPLE (IF ONE IS IN PROGRESS) TO TIME OUT IN CASE INITIALIZATION IS THE RESULT OF A POWER FAILURE. THE TIMEOUT WILL OCCUR IN 0-1 SECONDS.

THE DATA BUFFER REGISTER IS READ TO CLEAR A PENDING CONVERSION, INTERRUPT, OR ERROR FOR DEVICE INITIALIZATION.

INPUTS:

R4 = AD11 CSR ADDRESS
R5 = IDB ADDRESS OF DEVICE UNIT
R6 = ADDR OF DDB
R8 = ADDR OF CRB

OUTPUTS:

ALL REGISTERS PRESERVED

AD_CTLINIT:

CLRW AD_CSR(R4)
TSTW AD_DBR(R4)
RSB

: CLEAR CSR (IE IN PARTICULAR)
: CLEAR ANY PENDING CONVERSION,
: INTERRUPT, OR ERROR
:


```
.SBTTL AD_UNITINIT: AD11-K UNIT INITIALIZATION
```

```
AD_UNITINIT - AD11-K UNIT INITIALIZATION
```

```
THIS ROUTINE IS CALLED AT SYSTEM STARTUP AND AFTER A POWER  
FAILURE. THE UCB AND IDB ARE INITIALIZED.
```

```
INPUTS:
```

```
R5 = ADDRESS OF DEVICE UCB
```

```
OUTPUTS:
```

```
R0 = IDB ADDRESS  
OTHER REGISTERS ARE PRESERVED  
UCB$W_STS(R5), ONLINE BIT IS SET  
IDB$$_OWNER(R0) = ADDRESS OF OWNING UCB
```

```
AD_UNITINIT:
    BISW    #UCB$M_ONLINE,-      ;SET UNIT ONLINE
           UCB$W_STS(R5)
    MOVL    UCB$$_CRB(R5),R0      ;GET CRB ADDRESS
    MOVL    CRB$$_INTD+VE$$_IDB(R0),R0 ;GET IDB ADDR
    MOVL    R5,IDB$$_OWNER(R0)   ;SET UCB ADDR OF OWNING UNIT
    RSB
AD_END:
           ;END OF DRIVER LABEL
```

```
.END
```


0157	0158	0159	0160	0161	0162	0163	0164	0165	0166	0167	0168	0169	0170	0171	0172	0173	0174	0175	0176	0177	0178	0179	0180	0181	0182	0183	0184	0185	0186	0187	0188	0189	0190	0191	0192	0193	0194	0195	0196	0197	0198	0199	0200	0201	0202	0203	0204	0205	0206	0207	0208	0209	0210	0211	0212	0213	0214	0215	0216	0217	0218	0219	0220	0221	0222	0223	0224	0225	0226	0227	0228	0229	0230	0231	0232	0233	0234	0235	0236	0237	0238	0239	0240	0241	0242	0243	0244	0245	0246	0247	0248	0249	0250	0251	0252	0253	0254	0255	0256	0257	0258	0259	0260	0261	0262	0263	0264	0265	0266	0267	0268	0269	0270	0271	0272	0273	0274	0275	0276	0277	0278	0279	0280	0281	0282	0283	0284	0285	0286	0287	0288	0289	0290	0291	0292	0293	0294	0295	0296	0297	0298	0299	0300	0301	0302	0303	0304	0305	0306	0307	0308	0309	0310	0311	0312	0313	0314	0315	0316	0317	0318	0319	0320	0321	0322	0323	0324	0325	0326	0327	0328	0329	0330	0331	0332	0333	0334	0335	0336	0337	0338	0339	0340	0341	0342	0343	0344	0345	0346	0347	0348	0349	0350	0351	0352	0353	0354	0355	0356	0357	0358	0359	0360	0361	0362	0363	0364	0365	0366	0367	0368	0369	0370	0371	0372	0373	0374	0375	0376	0377	0378	0379	0380	0381	0382	0383	0384	0385	0386	0387	0388	0389	0390	0391	0392	0393	0394	0395	0396	0397	0398	0399	0400	0401	0402	0403	0404	0405	0406	0407	0408	0409	0410	0411	0412	0413	0414	0415	0416	0417	0418	0419	0420	0421	0422	0423	0424	0425	0426	0427	0428	0429	0430	0431	0432	0433	0434	0435	0436	0437	0438	0439	0440	0441	0442	0443	0444	0445	0446	0447	0448	0449	0450	0451	0452	0453	0454	0455	0456	0457	0458	0459	0460	0461	0462	0463	0464	0465	0466	0467	0468	0469	0470	0471	0472	0473	0474	0475	0476	0477	0478	0479	0480	0481	0482	0483	0484	0485	0486	0487	0488	0489	0490	0491	0492	0493	0494	0495	0496	0497	0498	0499	0500	0501	0502	0503	0504	0505	0506	0507	0508	0509	0510	0511	0512	0513	0514	0515	0516	0517	0518	0519	0520	0521	0522	0523	0524	0525	0526	0527	0528	0529	0530	0531	0532	0533	0534	0535	0536	0537	0538	0539	0540	0541	0542	0543	0544	0545	0546	0547	0548	0549	0550	0551	0552	0553	0554	0555	0556	0557	0558	0559	0560	0561	0562	0563	0564	0565	0566	0567	0568	0569	0570	0571	0572	0573	0574	0575	0576	0577	0578	0579	0580	0581	0582	0583	0584	0585	0586	0587	0588	0589	0590	0591	0592	0593	0594	0595	0596	0597	0598	0599	0600	0601	0602	0603	0604	0605	0606	0607	0608	0609	0610	0611	0612	0613	0614	0615	0616	0617	0618	0619	0620	0621	0622	0623	0624	0625	0626	0627	0628	0629	0630	0631	0632	0633	0634	0635	0636	0637	0638	0639	0640	0641	0642	0643	0644	0645	0646	0647	0648	0649	0650	0651	0652	0653	0654	0655	0656	0657	0658	0659	0660	0661	0662	0663	0664	0665	0666	0667	0668	0669	0670	0671	0672	0673	0674	0675	0676	0677	0678	0679	0680	0681	0682	0683	0684	0685	0686	0687	0688	0689	0690	0691	0692	0693	0694	0695	0696	0697	0698	0699	0700	0701	0702	0703	0704	0705	0706	0707	0708	0709	0710	0711	0712	0713	0714	0715	0716	0717	0718	0719	0720	0721	0722	0723	0724	0725	0726	0727	0728	0729	0730	0731	0732	0733	0734	0735	0736	0737	0738	0739	0740	0741	0742	0743	0744	0745	0746	0747	0748	0749	0750	0751	0752	0753	0754	0755	0756	0757	0758	0759	0760	0761	0762	0763	0764	0765	0766	0767	0768	0769	0770	0771	0772	0773	0774	0775	0776	0777	0778	0779	0780	0781	0782	0783	0784	0785	0786	0787	0788	0789	0790	0791	0792	0793	0794	0795	0796	0797	0798	0799	0800	0801	0802	0803	0804	0805	0806	0807	0808	0809	0810	0811	0812	0813	0814	0815	0816	0817	0818	0819	0820	0821	0822	0823	0824	0825	0826	0827	0828	0829	0830	0831	0832	0833	0834	0835	0836	0837	0838	0839	0840	0841	0842	0843	0844	0845	0846	0847	0848	0849	0850	0851	0852	0853	0854	0855	0856	0857	0858	0859	0860	0861	0862	0863	0864	0865	0866	0867	0868	0869	0870	0871	0872	0873	0874	0875	0876	0877	0878	0879	0880	0881	0882	0883	0884	0885	0886	0887	0888	0889	0890	0891	0892	0893	0894	0895	0896	0897	0898	0899	0900	0901	0902	0903	0904	0905	0906	0907	0908	0909	0910	0911	0912	0913	0914	0915	0916	0917	0918	0919	0920	0921	0922	0923	0924	0925	0926	0927	0928	0929	0930	0931	0932	0933	0934	0935	0936	0937	0938	0939	0940	0941	0942	0943	0944	0945	0946	0947	0948	0949	0950	0951	0952	0953	0954	0955	0956	0957	0958	0959	0960	0961	0962	0963	0964	0965	0966	0967	0968	0969	0970	0971	0972	0973	0974	0975	0976	0977	0978	0979	0980	0981	0982	0983	0984	0985	0986	0987	0988	0989	0990	0991	0992	0993	0994	0995	0996	0997	0998	0999	1000	1001	1002	1003	1004	1005	1006	1007	1008	1009	1010	1011	1012	1013	1014	1015	1016	1017	1018	1019	1020	1021	1022	1023	1024	1025	1026	1027	1028	1029	1030	1031	1032	1033	1034	1035	1036	1037	1038	1039	1040	1041	1042	1043	1044	1045	1046	1047	1048	1049	1050	1051	1052	1053	1054	1055	1056	1057	1058	1059	1060	1061	1062	1063	1064	1065	1066	1067	1068	1069	1070	1071	1072	1073	1074	1075	1076	1077	1078	1079	1080	1081	1082	1083	1084	1085	1086	1087	1088	1089	1090	1091	1092	1093	1094	1095	1096	1097	1098	1099	1100	1101	1102	1103	1104	1105	1106	1107	1108	1109	1110	1111	1112	1113	1114	1115	1116	1117	1118	1119	1120	1121	1122	1123	1124	1125	1126	1127	1128	1129	1130	1131	1132	1133	1134	1135	1136	1137	1138	1139	1140	1141	1142	1143	1144	1145	1146	1147	1148	1149	1150	1151	1152	1153	1154	1155	1156	1157	1158	1159	1160	1161	1162	1163	1164	1165	1166	1167	1168	1169	1170	1171	1172	1173	1174	1175	1176	1177	1178	1179	1180	1181	1182	1183	1184	1185	1186	1187	1188	1189	1190	1191	1192	1193	1194	1195	1196	1197	1198	1199	1200	1201	1202	1203	1204	1205	1206	1207	1208	1209	1210	1211	1212	1213	1214	1215	1216	1217	1218	1219	1220	1221	1222	1223	1224	1225	1226	1227	1228	1229	1230	1231	1232	1233	1234	1235	1236	1237	1238	1239	1240	1241	1242	1243	1244	1245	1246	1247	1248	1249	1250	1251	1252	1253	1254	1255	1256	1257	1258	1259	1260	1261	1262	1263	1264	1265	1266	1267	1268	1269	1270	1271	1272	1273	1274	1275	1276	1277	1278	1279	1280	1281	1282	1283	1284	1285	1286	1287	1288	1289	1290	1291	1292	1293	1294	1295	1296	1297	1298	1299	1300	1301	1302	1303	1304	1305	1306	1307	1308	1309	1310	1311	1312	1313	1314	1315	1316	1317	1318	1319	1320	1321	1322	1323	1324	1325	1326	1327	1328	1329	1330	1331	1332	1333	1334	1335	1336	1337	1338	1339	1340	1341	1342	1343	1344	1345	1346	1347	1348	1349	1350	1351	1352	1353	1354	1355	1356	1357	1358	1359	1360	1361	1362	1363	1364	1365	1366	1367	1368	1369	1370	1371	1372	1373	1374	1375	1376	1377	1378	1379	1380	1381	1382	1383	1384	1385	1386	1387	1388	1389	1390	1391	1392	1393	1394	1395	1396	1397	1398	1399	1400	1401	1402	1403	1404	1405	1406	1407	1408	1409	1410	1411	1412	1413	1414	1415	1416	1417	1418	1419	1420	1421	1422	1423	1424	1425	1426	1427	1428	1429	1430	1431	1432	1433	1434	1435	1436	1437	1438	1439	1440	1441	1442	1443	1444	1445	1446	1447	1448	1449	1450	1451	1452	1453	1454	1455	1456	1457	1458	1459	1460	1461	1462	1463	1464	1465	1466	1467	1468	1469	1470	1471	1472	1473	1474	1475	1476	1477	1478	1479	1480	1481	1482	1483	1484	1485	1486	1487	1488	1489	1490	1491	1492	1493	1494	1495	1496	1497	1498	1499	1500	1501	1502	1503	1504	1505	1506	1507	1508	1509	1510	1511	1512	1513	1514	1515	1516	1517	1518</
------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	------	--------